The Capacity of Color Histogram Indexing

Markus Stricker

Communications Technology Laboratory
Swiss Federal Institute of Technology ETH
CH-8092 Zürich, Switzerland

Michael Swain

Department of Computer Science
The University of Chicago
Chicago, IL 60637

Abstract

Color histogram matching has been shown to be a promising way of quickly indexing into a large image database. Yet, few experiments have been done to test the method on truly large databases, and even if they were performed, they would give little guidance to a user wondering if the technique would be useful with his or her database. In this paper we define and analyze a measure relevant to extending color histogram indexing to large databases: capacity *(how many distinguishable histograms can be stored).*

1 Introduction

As the cost of data storage drops dramatically, image databases are growing in size; soon many image databases will contain tens or hundreds of thousands of images. The labor involved with cataloguing images by hand, and the difficulty of anticipating every user's needs when assigning keywords to images, has led to the development of algorithms for retrieving images by their content. The goal of these algorithms is to quickly retrieve the images that are similar to a given image, or user-created image representation (*e.g.* a color histogram). The user may be looking for an image he or she has seen before, for another image of the same object or scene, or for an image that is similar along some dimensions to one that was liked. Potential applications include stock photo databases for electronic publishing, consumers searching their digital image databases created from such technologies as Photo-CD.

Typically, one feature vector per image is stored as an index for the database. A metric on the feature space is then used to retrieve images. Given an image, the distances between its feature vector and the feature vectors in the index are computed. Images for which this distance is less than a predefined threshold are subsequently retrieved.

Several researchers have proposed to use different kinds of color histograms as the features vectors to be stored in the index ([Swain and Ballard 1991, Funt and Finlayson 1991, Stricker 1992, Swain 1993]). In the QIBC system ([Niblack *et al.* 1993]) color, texture and shape are used to build an index. Feature vectors based on edge properties and texturedness of the images were proposed by [Nelson 1991, Engelson and McDermott 1991]. All the above mentioned papers report good results, but none of the algorithms has been tested on more than a 1000 images, *i.e.*, a truly large image database.

In this paper we present a careful theoretical analysis of one indexing technique. We compare the theoretical results with the results obtained from a Monte Carlo simulation of a large image database and with the data extracted from the Smithsonian Image Database[1]. Since many of the known techniques use various kinds of color histograms, we decided to focus on color histogram indexing. But the principles of our analysis should apply to other indexing algorithms based on feature vectors.

In the next section we briefly describe the indexing algorithm which will be analyzed in the following sections. In section 3 we discuss the metrical properties of the histogram space. This will reveal a basic restriction for indexing algorithms that use color histograms as their indices. In the following section we derive a lower bound for the capacity of a histogram space which is the maximal number of models that fit into the feature space. The proof of our results expose an interesting connection between binary coding theory and the histogram space. In the last section we indicate a future direction of research.

[1]The Smithsonian image database is a publicly accessible image database (ftp site photo1.si.edu) that contains approximately 500 color images. Museum items and photographs of outdoor scenes make up a large part of the images.

2 The indexing algorithm

The histogram matching algorithm we analyze in this paper is essentially the same as those presented in [Swain and Ballard 1991, Funt and Finlayson 1991, Stricker 1992].

We map the colors in an image into a discrete color space containing n colors. A color histogram $H(M)$ is a vector $(h_1, h_2, \ldots, h_n)$ in a n-dimensional vector space, where each element h_j represents the number of pixels of color j in the image M. We assume that all images have been scaled to contain the same number of pixels N before histogramming. These histograms are the feature vectors to be stored as the index of the image database. We will refer to the images of the database as the *models*.

To measure the distance d between two histograms H and I one can use the metric induced by the L_1-norm as in [Swain and Ballard 1991, Funt and Finlayson 1991, Stricker 1992] or a metric which is similar to the one induced by the L_2-norm (see [Niblack *et al.* 1993]). For the L_1-norm the distance is defined as

$$d_{L_1}(I, H) = ||I - H||_{L_1} = \sum_{l=1}^{n} |i_l - h_l|$$

and for the L_2-norm it is

$$d_{L_2}(I, H) = ||I - H||_{L_2} = \sqrt{\sum_{l=1}^{n} (i_l - h_l)^2}$$

The L_1-distance between two histograms is always less than twice the number of pixels per image and the L_2-distance is less than $\sqrt{2}$ times the number of pixels per image. For a given distance t, we say that two histograms are *t-similar* if their distance is less than or equal to t and *t-different* if their distance is greater than t. Now we can formulate the indexing algorithm concisely: For a fixed retrieval threshold t, a model is going to be retrieved if its histogram is t-similar to the histogram I of another image presented to the system.

3 The histogram space

Before we can study the algorithm we need to investigate the metrical properties of the histogram space. This will unveil a basic restriction of color histograms as indices of an image database.

Since all the images were scaled to contain the same number N of pixels, the histogram space $\mathcal{H}$ is the following subset of an n-dimensional vector space:

$$\mathcal{H} = \{(h_1, h_2, \ldots, h_n) \mid h_i \geq 0 \; (1 \leq i \leq n), \sum_{i=1}^{n} h_i = N\}$$

Recall that every simplex can be decomposed into simplices of lower dimensions, *i.e.*, the faces of the original simplex. It follows from the above equation that the histogram space is a face of a n-dimensional simplex and thus it is a $(n-1)$-dimensional simplex.

In order for color histogram indexing to work, the distance between histograms of completely different images must be large, *i.e.*, their histograms must be t-different for some distance threshold t. To determine the interval of reasonable values for t we study the distance distribution of the color histograms obtained from the Smithsonian image database[2] and from a randomly generated database. Figure 1 displays these distributions for the L_1-metric. The procedure that generates random color histograms is described in the appendix. Although we made no attempt to model the Smithsonian image database with our random color histograms, the distance distributions of both databases have qualitatively similar features: Very few histograms are very close together. The distributions have two very pronounced modes of which one is at the maximal distance. The location of the first mode is not fixed. Many of the museum items in the Smithsonian database were photographed in front of the same background and thus their histograms are relatively close together. As a consequence the first mode of the distance distribution for the Smithsonian database occurs at a smaller distance than the one for the randomly generated database. In general, the location of the first mode depends on the color composition of the domain from which the images were taken. The distance between two color histograms is maximal if and only if the intersection of the non-empty bins of the two histograms is empty. Thus, the mode at the maximal distance is produced by the sparseness of the histograms.

If the threshold t is larger than the location of the first mode of the distance distribution, then the indexing algorithm produces too many mismatches. If t is too close to 0, then the indexing procedure is too strict. Thus, the interval of reasonable values for t coincides with the first interval on which the distribution increases very rapidly. In Figure 1 this interval stretches from 25% to 60% of the maximal distance for the Smithsonian image database and from

[2] Prior to histogramming, we cropped any borders on the images, and smoothed and scaled the images to contain 10 000 pixels.

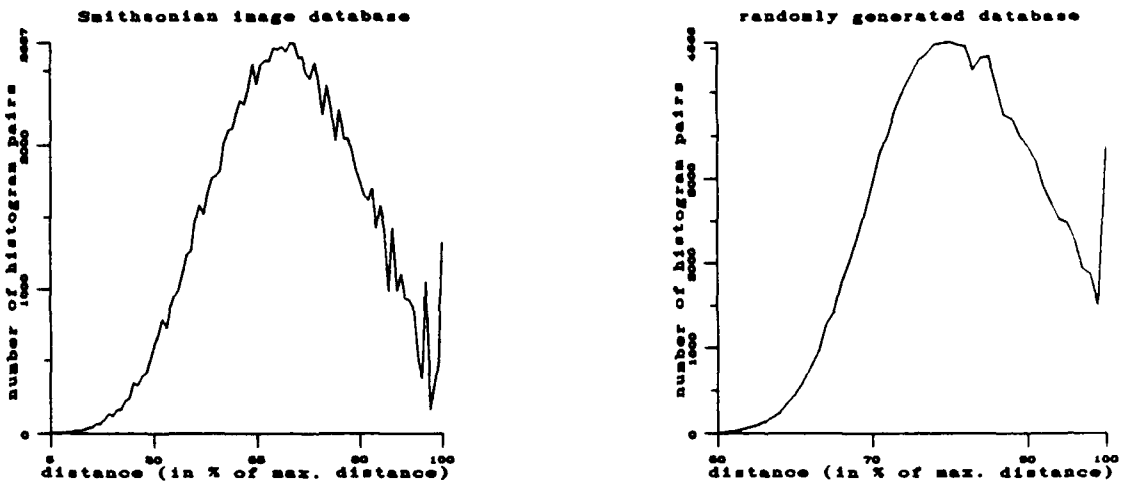

Figure 1: Distance distributions of 500 color histograms in the L_1-metric.

50% to 75% of the maximal distance for the randomly generated database. The interval for the Smithsonian image database contains the thresholds that were reported to work well in [Swain and Ballard 1991, Stricker 1992].

Observation 1 *The interval of reasonable values for t coincides with the first interval on which the distance distribution increases very rapidly.*

Note that in order to get a qualitative impression of the distance distribution it suffices to look at a small, random sample of the images in the image database.

In the following we analyze how so many large distances can be realized in the histogram space $\mathcal{H}$. Consider the barycenter $(\frac{N}{n}, \frac{N}{n}, \ldots, \frac{N}{n})$ of the histogram space. This vector corresponds to the histogram of an image with equal amounts of all colors. The barycenter is close to almost all the histogram vectors in $\mathcal{H}$. The only histograms that are far away from the barycenter are those that lie close to a lower dimensional face of the histogram space. In general, large distances between histograms can only be achieved if the histograms are in sufficiently different faces of the histogram space. In terms of colors and images this means that images containing the same colors or images containing all the colors of the color space are likely to have histograms that are close together.

Observation 2 *Indexing by color histograms works only if the histograms are sparse, i.e., most of the images contain only a fraction of the number of colors of the color space.*

From now on we assume that all the histograms are sparse.

4 The capacity of a histogram space

Before building an image database index one would like to determine whether color histograms provide a good tool to distinguish between many different images from a given environment. The capacity of a histogram space that we study in this section addresses this problem. We investigate the maximal number of different models in a database that can be retrieved without confusion by the indexing algorithm. This is the maximal number of t-different histograms that fit into a histogram space. We will derive a lower bound for this number. The proof of our result shows a very interesting connection between histograms and coding theory. The reader who is not familiar with the basic concepts of coding theory, such as the Hamming distance, can find them in [van Lint 1992].

Let us start with a formal definition of the capacity.

Definition 3 *Given a n-dimensional histogram space $\mathcal{H}$, a metric d on $\mathcal{H}$ and a distance threshold t, the capacity of $\mathcal{H}$ is defined as the maximal number of t-different histograms that fit into $\mathcal{H}$. We denote the capacity by $C(\mathcal{H}, d, t)$.*

The capacity also depends on the distribution of the colors in the images. It is very hard to account for an arbitrary color distribution and hence we study

the case where the distribution is uniform across the color space.

In [Swain and Ballard 1991] the authors propose to measure the capacity by dividing the volume of the histogram space by the volume of a $(n-1)$-dimensional sphere of radius $\frac{t}{2}$. The centers of the spheres correspond to the histograms. This simple sphere packing technique does not capture the capacity because it assumes that the whole spheres lie inside the histogram space, where in fact only the centers of the spheres (*i.e.*, the histograms) have to lie in $\mathcal{H}$.

The following theorem connects the capacity of a histogram space to a number which is well studied in coding theory. The statement that such a connection exists is by itself a novel and interesting remark.

Theorem 4 *Let $A(n, 2l, w)$ denote the maximal number of codewords in any binary code of length n, constant weight w and Hamming distance $2l$. Then the capacity of the histogram space $\mathcal{H}$ satisfies the following inequality:*

$$C(\mathcal{H}, d, t) \geq \max_{\substack{w,l \\ \alpha \leq l \leq w \leq n \\ l \leq n/2}} A(n, 2l, w)$$

where α is defined as $\frac{wt}{2N}$ in the L_1-case and as $\frac{1}{2}\left(\frac{wt}{N}\right)^2$ in the L_2-case.

Proof: We have already noted in section 3 that large distances between histograms can only be achieved if the histograms are in sufficiently different faces of the histogram space. For each face of $\mathcal{H}$ we represent the histogram at the barycenter of this face by the vector $(0, 1, \ldots, 1, 0, \ldots, 1, 0)$, *i.e.*, a binary word of length n with exactly w 1's in it. Obviously, $w \leq n$. Now we have to translate the condition that the histograms have to be t-different into a condition on binary words. A binary word with w 1's corresponds to a histogram $(0, \ldots, \frac{N}{w}, \ldots, \frac{N}{w}, 0, \ldots)$ with w entries $\frac{N}{w}$. Let $2l$ be the number of bins in which two such histograms H_1 and H_2 differ. Obviously l has to be less than or equal to w. The distance between H_1 and H_2 is

$$d_{L_1} = ||H_1 - H_2||_{L_1} = 2l\left(\frac{N}{w}\right)$$

and

$$d_{L_2} = ||H_1 - H_2||_{L_2} = \sqrt{2l}\left(\frac{N}{w}\right) .$$

If the inequalities $d_{L_1} \geq t$ and $d_{L_2} \geq t$ are solved for l, then they yield the values for α. Thus binary codewords of length n, weight w and Hamming distance $2l$, with l satisfying the stated inequalities, correspond to t-different histograms at the barycenters of the faces of $\mathcal{H}$. Since these are not completely general histograms, the maximal number of these histograms is smaller than the capacity. This is true for any admissable pair of parameters w and l. Thus, the maximum of $A(n, 2l, w)$ over l and w is still smaller than the capacity which completes the proof. □

For completely general values of l and w the number $A(n, 2l, w)$ is not known exactly. Thus, in most cases it is impossible to determine the maximum in theorem 4. But there exists a rich literature on bounds for $A(n, 2l, w)$ (see for example [van Lint 1992, Best *et al.* 1978, Graham and Sloan 1980]). We use a lower bound for $A(n, 2l, w)$ from [Graham and Sloan 1980] and theorem 4 to derive an explicitly computable lower bound for the capacity.

Corollary 5 *Let q be the smallest prime power such that $q \geq n$. In the L_1-case we set $l(w) = \lceil \frac{wt}{2N} \rceil$ and $\beta = n$, and in the L_2-case we set $l(w) = \lceil \frac{1}{2}\left(\frac{wt}{N}\right)^2 \rceil$ and $\beta = \min(n, \lfloor 2\left(\frac{N}{t}\right)^2 \rfloor)$. Then the capacity satisfies the following inequality:*

$$C(\mathcal{H}, d, t) \geq \max_{w \leq \beta} \frac{1}{q^{l(w)-1}} \begin{pmatrix} n \\ w \end{pmatrix}$$

Proof: From its meaning in coding theory it is clear that the maximum in theorem 4 is attained for the minimal value of l, *i.e.*, $l = \lceil \alpha \rceil$. [Graham and Sloan 1980, page 39] contains a short discussion on which lower bounds of $A(n, 2l, w)$ provide the tightest approximation on which range of parameters. Based on this discussion we choose [Graham and Sloan 1980, theorem 4] to establish the assertion of the corollary. □

To expose the strengths and the weaknesses of our lower bound, we compare its value with the capacity found by using a Monte Carlo algorithm. The details of how the Monte Carlo algorithm generates random color histograms can be found in the appendix. Table 1 displays the data for this comparison. Our lower bound is tighter for larger values of the threshold t. This stems from the coding theory bound that we used to prove our result. Tighter approximations for small values of t can be obtained in same way as we have shown by using [Graham and Sloan 1980, theorem 7]. For small values of t the capacity of the histogram space is so large that we normally do not have to worry about "overfilling" the histogram space.

t in % of max. distance	capacity	
	lower bound	Monte Carlo
50%	18.304	> 25,000
55%	2369	9829
60%	1861	2394
65%	651	681

Table 1: Capacity values for the L_1 induced metric on a histogram space with 64 colors. The Monte Carlo algorithm which is used to create histograms is described in the appendix.

Thus, we have chosen on purpose a method that provides a tighter bound in the more critical range of the capacity.

5 Future work

A large capacity is an indicator that many essentially different color histograms can be stored in the histogram space. The capacity does not measure how successful the indexing algorithm can retrieve them. In a forthcoming paper we will study how the retrieval performance of an indexing algorithm can be predicted using a small number of test images.

Appendix: Random color histograms

Realistic random color histograms can be generated with the following procedure. We assume that the number of non-zero bins per histogram is uniformly distributed between $c_{\min}$ and the dimension of the color space. Since color histogram indexing was conceived for multicolored images, we may exclude two-colored images and thus assume that $c_{\min}$ is 3. We use a color space of dimension 64. For each histogram we choose the number of non-zero bins c. Then we choose the colors of these non-zero bins. And finally, we generate the bin counters with a multinomial distribution on the c bins with probability $\frac{1}{c}$ for each color. The distance distribution of a database containing histograms generated with this procedure is shown in Figure 1.

Acknowledgment

A part of this work was carried out while the first author was working at the Computer Science Department at the University of Chicago.

References

[Best *et al.* 1978] M. R. Best, A. E. Brouwer, et al. Bounds for binary codes of length less than 25. *IEEE Trans. on Information Theory*, 24(1):81 93, January 1978.

[Engelson and McDermott 1991] S. P. Engelson and D. V. McDermott. Image signatures for place recognition and map construction. In *SPIE Proc. Sensor Fusion IV: Control Paradigms and Data Structures*, pages 282–293, 1991.

[Funt and Finlayson 1991] B. V. Funt and G. D. Finlayson. Color constant color indexing. Technical report, School of Computing Science, Simon Fraser University, Vancouver, B.C., Canada, 1991.

[Graham and Sloan 1980] R. L. Graham and J. A. Sloan. Lower bounds for constant weight codes. *IEEE Trans. on Information Theory*, 26(1):37–43, Jan. 1980.

[Nelson 1991] R. C. Nelson. Visual homing using an associative memory. *Biological Cybernetics*, 65:281–291, 1991.

[Niblack *et al.* 1993] W. Niblack, R. Barber, et al. The QIBC project: Querying images by content using color, texture and shape. In *IS&T/SPIE International Symposium on Electronic Imaging: Science & Technology, Conference 1908, Storage and Retrieval for Image and Video Databases*, Feb. 1993.

[Stricker 1992] M. A. Stricker. Color and geometry as cues for indexing. Technical report, Department of Computer Science, The University of Chicago, Nov. 1992.

[Swain and Ballard 1991] M. J. Swain and D. H. Ballard. Color indexing. *Intern. Journal of Computer Vision*, 7(1):11–32, 1991.

[Swain 1993] Michael J. Swain. Interactive indexing into image databases. In *IS&T/SPIE International Symposium on Electronic Imaging: Storage and Retrieval for Image and Video Databases*, pages 95–103, Feb. 1993.

[van Lint 1992] J. H. van Lint. *Introduction to Coding Theory*. Graduate Texts in Mathematics. Springer-Verlag, 2 edition, 1992.